PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 17/30 — A1

(11) International Publication Number: WO 99/41684

(43) International Publication Date: 19 August 1999 (19.08.99)

(21) International Application Number: PCT/US99/03028

(22) International Filing Date: 11 February 1999 (11.02.99)

(30) Priority Data:

60/104,597	13 February 1998 (13.02.98)	US
09/023,576	13 February 1998 (13.02.98)	US

(71) Applicant: FAST TV [US/US]; Suite 1550, 5670 Wilshire Boulevard, Los Angeles, CA 90036 (US).

(72) Inventors: KAZEROONIAN, Ali, S.; 328 Halyard Lane, Foster City, CA 94404 (US). KAMINS, David; 55 Highledge Avenue, Wellesley, MA 02181 (US). TWITCHELL, James, Pratt; 10 Arrowhead Circle, Chelmsford, MA 01824 (US). GAUCH, John, M.; 1101 Oak Tree Drive, Lawrence, KS 66049 (US). GAUCH, Susan, E.; 1101 Oak Tree Drive, Lawrence, KS 66049 (US). PANKRATZ, David, James; 3605 Stearns Hill Road, Waltham, MA 02154 (US). O'CONNELL, Robert, James; 16A Winslow Road, Brookline, MA 02146 (US).

(74) Agent: WALPERT, Gary, A.; Fish & Richardson P.C., 225 Franklin Street, Boston, MA 02110-2804 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

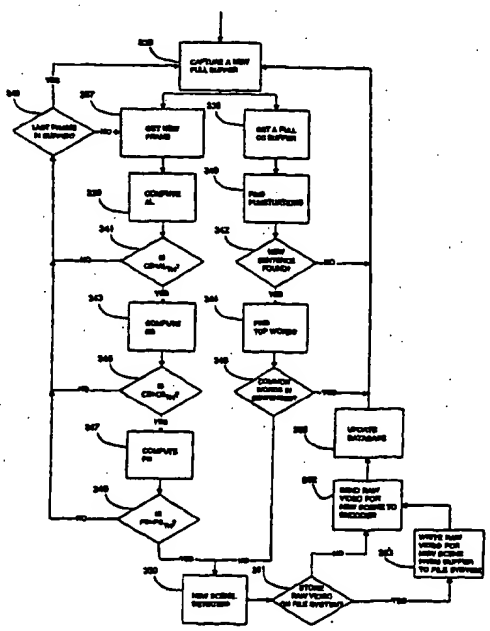
With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: PROCESSING AND DELIVERY OF AUDIO-VIDEO INFORMATION

(57) Abstract

An automated real-time system for processing and distribution of audio-video data. The system performs automated, real-time analysis and scene detection, segmentation, indexing, and encoding of video for real-time presentation, browsing, or searching. By using automated real-time processing of audio-video data sources, the data is available to a user (a viewer) without a substantial delay that would be introduced by manual or off-line (batch oriented) automatic processing of the data. The processing is arranged in a pipelined process, reducing processing delays and requiring less intermediate storage than a batch-oriented processing approach. The audio-video sources are segmented into individual scenes, such as one story in a news broadcast, thereby allowing a user to access portions of source programming without having to view or scan through the entire programs or to specify a particular time interval. The system also makes combined use of video, audio, and closed-caption information in an audio-video signal to identify individual scenes. This combined use of multiple sources of information provides an improved scene detection capability. Characterization of individual scenes is based on a variety of sources of information including, for example, the closed-caption text and the output of a speech recognizer analyzing the audio information in the signal.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

PROCESSING AND DELIVERY OF AUDIO-VIDEO INFORMATION

Background

This invention relates to processing and delivery of audio-video information.

Audio-video information is broadcast to viewers, using a variety of communication media and techniques including, for example, conventional television transmissions, digital or analog satellite transmissions, or data transfer to networked personal computers (for example, through the Internet's World Wide Web). Limited experimental and non-commercial deployments of "interactive television" (ITV) and "video-on-demand" (VOD), for example by cable television providers, attempt to provide more targeted programming to viewers. These systems have not yet led to significant commercial viewership. Some of these interactive television systems offer the capability for a viewer to search a large pool of programs and to retrieve and view only a subset of programs that match his interest. Video-on-demand systems allow a user to select, retrieve and view one or more video programs in their entirety by selecting from an existing, pre-processed content menu. For example, the viewer may rely on the program title, a predetermined subject category, names of on-screen talent, and perhaps other tangential information to decide whether to select a particular program. A significant amount of time and resources are devoted to preparing the programs for availability on the video-on-demand selection menus.

Research systems that provide content-based access to audio-video libraries have also been developed. One such system, called "Informedia," has been developed by researchers at the Carnegie-Mellon University. This system incorporates mechanisms for detecting, indexing,

retrieving, and viewing video scenes in a stored audio-video library. The system requires the viewer to specify the keywords manually and then it retrieves the relevant videos. This and other systems use automatic methods for detecting scene changes. A variety of systems have also been developed for accessing audio-video information that has already been indexed using, for example, manual indexing techniques.

Audio-video processing techniques have been developed for detecting changes of scene based on video information. Such video scene detection has been used to allow a user to browse a video archive by viewing representative frames of each scene. Another use of automatic video scene detection has been to tag and index video for future retrieval, for example, to manage film and video assets. Systems with similar capabilities have also been based on image recognition techniques rather than video scene detection.

Summary of the Invention

The invention relates to an automated real-time system for the processing and distribution of audio-video data. One aspect of this system performs automated, real-time analysis and scene detection, segmentation, indexing, and encoding of video information for real-time presentation, browsing, or searching. By using automated real-time processing of audio-video data sources, the data is available to a user (a viewer) without a substantial delay that would be introduced by manual or off-line (batch oriented) automatic processing of the data. The processing is arranged in a pipelined process, reducing processing delays and requiring less intermediate storage than a batch-oriented processing approach. The audio-video sources are segmented into individual scenes, such as one story in a news broadcast,

thereby allowing a user to access portions of source programming without having to view or scan through entire programs or to specify particular time intervals. The system also makes combined use of video, audio, and closed-caption information in an audio-video signal to identify individual scenes. This combined use of multiple sources of information provides an improved scene detection capability. Characterization of individual scenes is based on a variety of sources of information including, for example, the closed-caption text and the output of a speech recognizer analyzing the audio information in the signal.

In one aspect, in general, the invention provides a method for fully automated real-time processing and delivery of an audio-video program. The method features accepting the audio-video program, for example from a satellite receiver, and detecting discrete scenes within the program based on the content of the program. For each of the discrete scenes, the method includes determining textual information related to the scene, indexing the scene using the textual information, optionally compressing or encoding the video data, storing audio-video data from the scene, and storing index data for the scene. The method can also include accepting the description of interests of a user, such as keywords or category names associated with topics of interest to that user. The method then includes matching the description of interests of a user to stored index data for the scenes, and providing audio-video data from the matching scenes to the user, for example over a data communication network.

The invention can include one or more of the following features.

Detecting scenes, determining textual information related to the scene, indexing the scenes, and storing

data from the scenes, can all occur in a pipelined manner while accepting the program. Providing audio-video data to the user can therefore begin prior to completion of accepting of the program. In this way, the user can view scenes of the program with low delay relative to accepting those scenes in the program.

The method can also include accepting a text document, for example from a text source such as a news source. For each of the discrete scenes, the method then includes further matching of the scene to the text document and, if a match is found, storing the matching information which associates the scene to the document. Providing audio-video data from the matching scenes then also includes providing any stored matching information which associates the scenes to the text document.

Scenes can be detected within the program based on the content of the program using both the audio and video portions of the program. This can include comparing a color distribution at one time in the program to the color distribution at another time, or computing a statistic of the rate of change of the video signal.

Scenes can also be detected by processing the closed captions of the program. This may involve, for example, comparing the frequency of previously selected words in one portion of the program to their frequency in another portion of the program. It may also involve detecting predetermined character sequences, such as punctuation or particular words, in the closed captions.

The textual information used for indexing can be determined by processing the closed captions of the scene. In addition, the information can be determined by processing the audio portion of the scene using an automatic speech recognizer.

If the audio-video data is sent to the user over a data network, it can be sent using a data streaming

protocol and the Internet Protocol (IP) network protocol. Also, it can be multicast for reception by multiple users. In addition, the data can be compressed in accordance with the communication capacity of a communication path over the data communication network to the user.

The description of interests of a user can be received prior to accepting an audio-video program, thereby forming a profile for that user. The description of interests can also be received after accepting the program, thereby forming a search request for that user.

In another aspect, in general, the invention features a system for fully automated real-time processing and delivery of an audio-video program. The system includes a segmenter/indexer for accepting the program and providing data for discrete scenes in the program, a media database for accepting and storing audio-video data for the discrete scenes from the segmenter/indexer, an information database for accepting and storing index data based on textual data related to the discrete scenes from the segmenter/indexer, and a communication network coupling the media database and a client computer for providing audio-video data stored in the media database which match a description of interests provided by a user of the client computer.

In another aspect, in general, the invention provides software stored on computer readable media for causing a computer implementation of such an audio-video processing and delivery system to function.

Other features and advantages of the invention will be apparent from the following description, and from the claims.

Description of the Drawings

Fig. 1 is a block diagram of physical components of a video processing system and its interconnections with data sources and users;

Fig. 2 is a block diagram of the software components of the audio-video processor, and the media and information databases;

Fig. 3 is a flowchart of operation of the audio-video processor;

Fig. 4 is a schematic of the pipeline stages of processing; and

Fig. 5 is a schematic of the time evolution of pipeline stages on a dual processor computer.

Description

Referring to Fig. 1, a user at client computer 109 requests data from a video processing system (VPS) 100, and accepts the requested data, which originated at audio-video sources 110 and text content sources 104. VPS 100 includes several server computers which process the audio-video and text data, respond to requests from users, and provide data to those users. An audio-video processing server (AVPS) 101 accepts data from audio-video sources 110, processes the audio-video data, and provides the processed data to a media server 102 where it is stored or buffered prior to being distributed to client computer 109. A text processing server (TPS) 105 accepts textual data, including text accompanying the audio-video data such as closed captioning (CC), as well as text data from other text content sources 104, such as news wire service, processes the text data, and provides processed data to media server computer 102. AVPS 101 and TPS 105 provide information related to the context and structure of the processed data to an information

server 103. Client computer 109 communicates with a Web server 106 in order to request audio-video data. Data stored in media server 102 is identified based on data stored in information server 103, and the identified audio-video and text data is sent from media server 102 to client computer 109 as a compressed data stream.

The server computers, AVPS 101, media server 102, information server 103, TPS 105, and Web server 106 are coupled to a local area network (LAN) 107. LAN 107 is coupled to a wide area network (WAN) 108, such as the Internet. Client computers 109 are also coupled to WAN 108. Each of the computers 109 includes a processor, working memory, a program storage, such as a fixed or removable magnetic disk drive, and a network communication interface. In addition, media server 102 and information server 103 include data storage devices, such as magnetic disk drives, and audio-video processing server 101 includes an additional processor enabling parallel processing on the audio-video server. LAN 107 provides reliable high-data-rate communication between the server computers coupled directly to the LAN, for example using Ethernet supporting communication rates in the range 10-100Mb/s. WAN 108 provides lower-rate and less reliable communication between LAN 107 and client computer 109. If a client computer is coupled to WAN 108 over a standard telephone modem, this rate may be only 28kb/s, while the rate may be as high as 1Mb/s or higher if the client is coupled to the WAN over a T1 telephone connection and the WAN provides as fast or faster path than LAN 107.

Audio-video sources 110 includes one or more real-time sources of audio-video data. The sources can include receivers of terrestrial or satellite broadcasts of analog or digitally-encoded television signals, or receivers of television signals provided over a wired

distribution system (for example, a cable television system) or over a data network (for example, the Internet). In addition, audio-video sources can include sources of non-real-time data, for example, recorded signals, such as edited television programming or feature films.

Audio-video data provided by audio-video sources 110 includes analog or digitally-encoded television signals. A television signal includes video and audio signals, as well as a closed-caption (CC) text signal, encoded during the vertical blanking interval of analog signals.

Turning now to the user's interface to the system, client computer 109 includes software that communicates with software on Web server 106 to search for data or to set a personal profile, and includes software that accepts an audio-video data stream originating at media server 102 and transported over WAN 108. In a search mode, the user searches the available archives of video clips and text. Once a list of desired video clips and news articles is compiled by the user, a corresponding data stream is transported from media server 102 to client computer 109. The data stream can be transported using a variety of protocols, such as ones using streaming techniques layered on the Internet Protocol (IP) network layer protocol. The data stream can be compressed to satisfy communication-rate constraints on the data path between media server 102 and client computer 109. Data can be transferred in a variety of ways from media server 102 to client computer 109, including using an on-demand (that is, direct) transmission, or as part of a broadcast or multicast transmission in which multiple client computers receive the same data streams. The client computer can also receive text data directly from text content sources 104

for concurrent presentation with audio-video data sent from media server 102.

Turning now to the server computers that make up VPS 100, AVPS 101 accepts a television signal from audio-video sources 110. Information related to changes in video (that is, from frame to frame), audio intensity, and punctuation and words in the CC text are used to split the video into distinct scenes (video clips). The video and audio from each scene is then compressed (encoded) and stored on media server 102. The CC text, as well as information about the particular video source, broadcast time, etc., for each scene is stored in information server 103 to enable future searches that might return those scenes.

TPS 105 receives CC text for each video scene directly from AVPS 101. TPS 105 assigns a category (a subject code or topic) to the CC text for a clip. This category information is provided to information server 103.

TPS 105 also receives text documents from text content sources 104, for example from a text server computer on site at a text news provider. The CC for an audio-video clip is used to find related documents from the text sources. The relationship information is provided to information server 103. An example of this type of correlation of sources is an audio-video clip of a weather forecast being matched to a textual weather forecast. This match could later be used to present to a user at client computer 109 the textual weather forecast side-by-side with the audio-video weather forecast broadcast. Based on this type of correlation, news stories that fall in the same category as a video clip can also be shown side by side. The user is therefore able to see retrieved video together with related news stories.

Server computer 106 includes server software that communicates with client software executing at client computer 109. For instance, a Web browser executes at client computer 109 and communicates with a Web server executing at server computer 106. The user, using the client software, provides the server software with a request for data, for example by filling out an electronic form for a keyword-based search or for selecting a category or topic. If Web client and server software is used, the form is transported to between server computer 106 and client computer 109 using the hyper-text transport protocol (http). The server software accepts the request, accesses information server 103, and compiles a list of video clips and news articles. This list is provided to the client software, in the form of a multimedia document or in some other form that can be interpreted by the client software. In response to input from the user, the client software then submits requests for the video clips to media server 102, and for text data to text content sources 104 and accepts the data provided in response to the requests.

A hyper-text list of video clips and news articles can include "thumbnail" views of video clips presented along with other identifying information for the clips. The thumbnail can be a static image, or can change while it is presented to the user. The frame chosen to be displayed can be chosen at the time the video clip is initially processed. For instance, the most stationary portion of the clip can be used to as the representative frame to display in the thumbnail view. A time-varying thumbnail can sample frames from the video clip, for example sampling uniformly in time, or using a previously chosen sequence of relatively stationary frames.

In an alternative mode of operation, rather than search for audio-video data that has already been stored

in the media database, the user can provide profile information in advance of the data being available. Then, as relevant audio-video data is processed, it can be provided to the user, for example, as a personalized broadcast.

Referring to Fig. 2, the details of a software audio-video processor 201 which executes on audio-video processing server (AVPS) 101 (shown in Fig. 1) is depicted. Each of the illustrated software modules executes on one or more of the processors in AVPS 101 and communicates with other modules in audio-video processor 201 through shared memory buffers in the working storage. Execution of the modules can be allocated to the multiple processors in a variety of ways, for example, with one processor digitizing while another is performing segmentation. No intermediate disk files of the digitized video are created prior to the data passing through a segmentation coordinator 217.

A digitizer/decoder 225 provides an interface for the audio-video data accepted from audio-video sources 110. In the case of analog television data, digitizer/decoder 225 controls the acquisition of the analog signal and extraction and digitization of video, audio, and closed-caption data in the signal. In the case of digitized data, digitizer/decoder decodes the digital data stream and creates digital streams for the video, audio, and closed-caption data. The digital video stream in both cases includes individual digitized frames. Each frame, as it is extracted, is stored in an indexed location in a video buffer 214. The corresponding digitized audio frame is stored in an audio buffer 223 at the same index location, thus insuring audio/video synchronization. The digitized closed-caption text is stored in a CC buffer 228. The correspondence between audio data and closed caption data

is preserved by keeping track of the number of bytes of CC data in a given video frame.

A video segmenter 213 performs two analyses on each frame. First, it computes a color histogram of the frame, and compares it with the histogram of the previous and the next frames to produce a video color difference quantity. This quantity is related to the rate of change of the image. This comparison is performed by summing the absolute differences of corresponding histogram values associated with different colors. In the second analysis, a pixel-to-pixel change comparison is made between the current frame and the previous frame by summing of absolute differences of all changes in the pixel intensity and color values to produce a pixel difference quantity. Both quantities computed by video segmenter 213 are passed to segmentation coordinator 217.

An audio segmenter 216 measures the average audio level for each frame to produce an audio level quantity. The audio level quantity is also passed to segmentation coordinator 217.

A CC segmenter 227 analyzes the CC data in CC Buffer 228 and checks for the presence of punctuation marks such as periods and for CC special characters such as carets (">>"). The CC segmenter also checks for the occurrence of common words across sentence boundaries. CC segmenter 227 passes a signal to segmentation coordinator 217 indicating whether a scene change is likely to have taken place based on the CC text. For example, the same important words used in two frames contributes to an indication that both frames are part of the same scene.

In segmentation coordinator 217, the video color difference, pixel difference, and audio level quantities are compared to respective thresholds stored in the segmentation coordinator. If all of the quantities

exceed their respective thresholds, the segmentation coordinator determines whether a scene boundary should be declared at that point. This declaration is based in part on the signal received from CC segmenter 227.

Once segmentation coordinator 217 determines that a scene boundary has occurred, the contents of video buffer 214, and audio buffer 223, including the audio-video data from the new scene, are passed to an encoder 231. Alternatively, the audio-video data can have been buffered in a temporary hard disk storage 234 and read by encoder 231 when the scene boundary is determined.

Encoder 231 compresses the video and audio data for a scene, and passes the compressed data to a media database 232, stored on media server 102. Media database 232 includes individual files for each of the scenes, as well as an index or directory to access those files.

Segmentation coordinator 217 also sends data for new entries to be stored in a content table 219 and a text table 220 in a database 218 stored on information server 103. The entry in content table 219 includes the location or index of the corresponding data stored in a media database 232, plus other information such as the duration of the scene and the name of the audio-video source. The entry in text table 220 contains all of the CC text for the given scene. The two entries are related by a common index so that once the database is searched for a specific word in the closed caption entries, all of the related video scenes locations can be found.

Segmentation coordinator 217 also passes the CC text for a scene to a text processor 240 executing on TPS 105. Text processor 240 matches the CC text to available text sources from text content sources 104.

The system can also include a speech recognizer. This speech recognizer takes, as an input, the audio portion of an audio-video program and produces a word

sequence corresponding to that audio data. This word sequence can be used to supplement, or can be used instead of, the CC text in the approach described above.

Referring to Fig. 3, in operation, the audio-video processor 201 determines whether each new frame of data begins a new scene. Audio-video processor 201 begins by capturing, in a buffer, multiple frames, for example a fixed duration interval, of the audio-video source data (step 336). This buffer is part of digitizer/decoder 225. One frame of data is then stored in each of video buffer 214 and audio buffer 223 (step 337). In parallel, the CC text for the entire buffer is stored in the CC buffer 228 (step 338). In many video sequences, speakers, anchor persons, or other talking persons stop their speech as the video subject matter changes from one topic to another. Video editors almost always paste together video segments with quiet audio tracks at both ends of the segment. Therefore, a quiet period in the audio track of a video is usually an indication of a possible video scene change. Audio Level quantity (AL) is measured (step 339) by taking the average of the audio level of the data in audio buffer 223. AL is then compared to a pre-set Audio Level Threshold (AL_{TH}) value (step 341). If the Audio Level is less than the Audio Level Threshold, then a quiet frame has been reached that satisfies the audio level condition for a scene boundary, and further consideration of this frame as a scene boundary continues. If the level condition is not satisfied, a new frame, if available, is processed (step 349). If the last frame of the buffer has been used then a new buffer is captured (step 336) otherwise the next frame from the buffer is read and processing continues (step 337).

Consideration of the frame continues by computing a histogram of the colors in the frame. In this

embodiment, an 8-bit color palette (that is, 256 colors or color classes) is used to compute the histogram. A video color difference quantity (color spectrum, CS) is determined as the sum of the absolute differences of the corresponding histogram values (step 343). The value of CS is compared to a threshold CS_{Th} (step 345) and if CS is greater than CS_{Th}, the processing determines that a significant change in the color of the frames has occurred (perhaps a new object is being depicted in the frame with new colors) suggesting the possible start of a new scene. If CS is not greater than CS_{Th}, then processing continues with the next frame (step 349).

Next the processing determines a pixel difference quantity (pixel spectrum, PS) (step 347). PS is determined as the sum of the absolute differences between each of the intensity (luminance) and color (chrominance) values of corresponding pixels in the current frame and the previous frame. The value of PS is compared to a threshold PS_{Th} (step 348) and if PS is greater than PS_{Th}, the processing concludes that a significant change in the whole frame has occurred, for example, caused by a movement of an object in the frame suggesting the possible start of a new scene. If PS is not greater than PS_{Th}, then processing continues with the next frame (step 349).

In parallel with the steps involving the audio-video frame, the text stored in the CC buffer (at step 338) is examined (step 340) to locate punctuation characters, including period (".), single caret (">), double carets (">>"), triple carets (">>>"), marking a new sentence, subject, or speaker. If a new sentence is found (step 342), then processing proceeds to identify the most important words in the CC text (step 344). Important words are identified by determining the frequency of their appearance in a representative textual

corpus. The least frequently occurring words are ranked as the most important words. The processing compares the top words in the present CC buffer to those of the previous buffer and if the two buffers have several words in common, then the processing decides that the subject matter of the two buffers are probably the same and the two buffers belong to the same scene (step 346) and processing proceeds with the next buffer.

If the current frame is a candidate scene change based on the audio and video signals (step 348) and the current buffer is a candidate scene change based on the CC text (step 346), then this frame is determined to be the first frame of a new scene (step 350).

If a new scene is found, the video and audio data buffered for the entire scene is indexed and compressed as follows.

Processing determines whether to store the buffers belonging to the present scene as raw video files on disk buffers rather than passing them directly to the encoder (step 351). If the disk storage mode is being used, then raw audio and video buffers for the new scene are written to the disk (step 353). Otherwise, the audio and video buffers are passed directly to the encoder (step 352).

Processing for the scene finishes with passing information about the scene to information database 218, including the CC text, the length of the scene, the publisher of the video, the broadcast time, and the encoding algorithm with which the audio and video data is encoded or compressed (step 355).

An additional criterion can be used to locate scene boundaries based on the video signal. The average intensity (luminosity) of each frame is compared to a threshold. When the intensity is below a threshold, a scene change is declared. This intensity threshold can be used to detect black frames that would usually

indicate a scene change during a broadcast. Most broadcasts fade their programming to black before a commercial or between unrelated shows.

An additional criterion can be used to locate scene boundaries based on the audio signal. A fall in the audio level quantity below a threshold for more than a minimum duration is used to detect a pause or lull in a speech in a scene, and a scene boundary can be declared at that point. For example, a short pause (lull) by a speaker can be construed as the end of a sentence and can be used for detecting a scene where accurate punctuation in closed caption can not be available. A lull duration of one second would indicate a possible scene boundary.

An additional criterion can be used to detect scene changes based on the CC text. Presence of a specified string or pattern can determine an automatic (overriding) boundary for scene. For example, when segmenting a video program covering the U.S. House of Representatives chamber discussions, automatic determination of a scene with every instance of the phrase "THE SPEAKER PRO TEMPORE:" is useful. This type of over-ride is most useful when a video signal is covering a live event or where very little action or movement is present (or likely to be) within the video stream itself.

Audio-video processing server (AVPS) 101 can be a multi-processor allowing parallel processing of the modules and tasks shown in Figs. 2 and 3. The tasks and processing steps are divided between separate processing threads or operating system processes. AVPS 101 uses threads to achieve real-time video encoding throughput, high segmentation speed, and other characteristics that affect user interface responsiveness. The pipelined architecture supported by these multiple threads provides

the end-users with access to the indexed videos only moments after the videos are received.

Even on a single processor system multi-threading can improve performance by judicious assignment of various tasks (stages) to separate threads. For example, as shown in Fig. 4, a video capture thread 458 is separate from a segmentation thread 459, allowing continuous (real-time) capture of video while the segmentation of the earlier video data is taking place.

Using threads is especially advantageous when a process consists of multiple tasks of varying urgency and processing needs. For example, the capture task (performed by the capture thread 458) is an urgent task: if the capture falls behind, the system loses video information that cannot be recovered. The encoding process (performed by an encoding and database update thread 461) requires significant processing power, especially if the video clips are being compressed to a high compression factor. Moreover, in some applications, it can be required that the same video content to be encoded to several different compression factors, or using different video file formats, simultaneously. In these cases and others, performance of the overall system is enhanced if encoding tasks are distinct from the capture and segmentation tasks. Note that more than one encoding and database update threads can be running at the same time.

Control of the processing threads is initiated by a (callback) message from the video capture driver 457 awakening capture thread 458 which begins processing. As soon as a new scene is found by segmentation thread 459, a new encoding and database update thread 461 is spawned to encode the corresponding video buffer data and update the database. Since segmentation can take less time than encoding, a single segmentation thread can spawn multiple

encoding and database update threads. Note that at any given time, there can be multiple video buffers being processed in thread pipeline 456.

Fig. 5 is a schematic of the time evolution of pipeline stages (tasks) in audio-video processor 201 on a dual-processor computer, according to one aspect of the invention. This schematic can be generalized to any number of processors.

As was shown previously in Fig. 4, different processor threads are assigned to different stages of processing. In Fig. 5, the vertical axis labeled stage 564 spans the three major stages of processing: video capture 565, video segmentation 566, and video encoding/database updates 567. For ease of discussion we assume that one of the processors (P_1) is assigned to handle the capture task. In other words, the capture thread runs on P_1 and handles the capture stage 565. The segmentation stage 566 is handled on threads running on either processor. Encoding and database stage 567 is handled by multiple threads running concurrently on the two processors. The horizontal axis represents time 570. The six time intervals marked on the horizontal axis represent the equal-length captured buffers and Δ represents the duration of each buffer. For the purposes of this discussion, the capture of the first buffer B_1 571 begins at time=0 (568) and ends at time=Δ(569). Similarly, the capture of the second buffer B_2 572 begins immediately after the completion of the capture of B_1. Each buffer B_1 contains, typically, a plurality of frames with video and related information.

Within a time interval Δ_G 576 after the start 573 of capture of B_1, the segmentation process S_1 577 begins (which results in a segment or a scene) for buffer B_1. Note that in general the duration of each segmentation process S_1, S_2, etc. is not the same as that of the

buffers B_1, B_2, . . . since the duration of each segmentation process is the same as the length of a scene, while buffers are all equal size. To make this difference clear, we extended the buffer boundary line 575 between B_1 and B_2 to cross S_1. Also note that the various segments S_1-S_6 (corresponding to 577-582) are typically of different lengths. In fact, as in S_1, a single segment can contain video frames from two (or more) buffers. Note also that S_1 can begin before the end of B_1 since the B_1 is a full buffer while the segmentation process takes place on each frame within a full buffer. The time offset Δ_G 576 is necessary because in many situations (especially when closed captioning is done in real-time by the video broadcasters) the closed caption text lags behind the video. Typically Δ_G is greater than this time lag.

At any given time there is only one segmentation thread: segmentation algorithms are not as latency sensitive as the capture process and they are not as computationally intensive as the encoding process. Depending on the details of the operating system scheduling, either of the two processors can be assigned to S_1, S_2, etc.

The encoding process E_1 558 begins a time Δ_E 587 after the start time 583 of S_1. $E_1(P2)$, which we assume is running on the first processor P_1, can take longer than the actual duration of the scene. This is because encoding is a time consuming process and P_1 can be performing multiple tasks. The gray projection 584 demonstrates the mapping between S_1 and the corresponding encoding process E_1.

Note, that the second segment (S_2 578) is ready for encoding E_2 before E_1 is completed. The pipeline architecture described here enables the audio-video processor to begin encoding the second segment before

encoding of the first segment is completed and thereby provide real-time scene detection, capture, and encoding. Note that E_1 and E_2 run concurrently for part of their lifetime (and in this case on different processors). Similarly, E_3 589 begins before the completion of E_2 and at certain times E_4 592, E_5 590, and E_6 593 all run concurrently.

The gray projection 585 shows the mapping between S_4 and E_4. Note that E_4 takes significantly longer than S_4, partly because E_4 is running on P_1 which is also occupied with the capture process. The gray projection 586 shows the mapping between S_6 582 and E_6 593.

Note that E_6 ends shortly after B_6, thus maintaining the throughput of the pipeline without any congestion.

It is understood to those skilled in the art of design of electronic and computer systems that different arrangements of the components are possible. For example, functions carried out by separate computers can be carried out on a single computer, and a single function can be distributed over multiple computers. For instance, the functions of media server 102 and information server 103 can be provided by a single computer. Also, information database 218 and media database 232 can be combined into a single database, and can use a variety of data storage approaches, for example, using a file system or an object oriented or a relational database. Other forms of data communication networks can be used, for example, using only a local area network. Also, client computers 109 can be diskless computers, receiving client software over a data network.

It is also understood that functions implemented by software executing a general purpose computer can also be implemented using a special purpose computer or other specialized hardware.

Finally, it is also understood that the foregoing description is intended to illustrate and not limit the scope of the invention, which is defined by the scope of the appended claims. Other aspects, advantages, and modifications are within the scope of the following claims.

What is claimed is: